COP4521 – Florida State Parks System Report 1

## Background

For the Florida State Parks Project, we have decided to use MySQL and host it on AWS, and we were planning to implement role-based access to manage user permissions. However, we encountered a problem while implementing this feature. The admin user provided by AWS during the initial setup did not have SUPER permissions, which prevented us from adding users to the groups we created.

## Group Creation

Initially, we created two groups: "park ranger" and "customer."

The park ranger group was supposed to have read, update, delete, and execute access, along with system access corresponding to park ranger.

Similarly, the customer group was meant to allow only read and update access and system access corresponding to the customer.

## SUPER Permissions Issue

We faced difficulty adding users to these groups because we did not have SUPER permissions in the database. We were hoping to create a user and assign them to a group, but unfortunately, that wasn't possible due to the lack of SUPER permissions. AWS doesn't provide SUPER permissions to the MySQL database by default for security reasons. SUPER permissions allow a user to perform tasks such as starting and stopping the database server, changing global system variables, and killing other users' sessions, and that is why AWS restricts those permissions.

## Workaround Solution

To tackle this issue, we came up with a workaround. We created a column called "isAdmin" on the Users table, which differentiated between park rangers and customers. We handled the isAdmin bit column on the Python side, where if the user was an admin, they would only have access to the admin pages, and if the user was a customer, they would only have access to the customer pages.

## Future Considerations

While this workaround worked for us, we acknowledge that it's not an ideal solution. In future projects, we plan to consider different hosts other than AWS to obtain SUPER permissions so that we can implement role-based access in a more streamlined manner.

## Conclusion

In conclusion, we were unable to implement role-based access as planned due to the lack of SUPER permissions provided by the AWS admin user. However, we found a workaround solution by creating a column on the Users table to differentiate between park rangers and customers. We plan to explore different hosting options in future projects to avoid this issue altogether.